

## MG51 Series BSP User Guide

Based on IAR Embedded Workbench 8051 V10.10.1

For NuMicro® 8051 Family

### Directory Information

Please extract the “MG51\_Series\_BSP\_IAR\_V1.00.000.zip” file firstly and confirm the following content of this BSP folder.

#### MG51

BSP for MG51 Series.

Up to 32KB Flash APROM share with 4KB LDROM

256 Byte RAM, 1024 Byte XRAM

In TSSOP20 & QFN20 package

Each folder listed above with following content folders

#### Document\

Driver reference manual and revision history.

#### Library\

Device driver header and source files.

#### SampleCode\

Driver sample code.

*The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.*

*Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.*

*All data and specifications are subject to change without notice.*

For additional information or questions, please contact: Nuvoton Technology Corporation.

[www.nuvoton.com](http://www.nuvoton.com)

## 1 .\Document\

Nuvoton\_MG51\_Series\_  
BSP\_IAR\_Revision\_Hist  
ory.pdf

This document shows the revision history of MG51 BSP for IAR Embedded Workbench for 8051.

## 2 .\Library\

Device\	MG51 Device header file
Startup\	A51 startup file and executable file
StdDriver\	All peripheral driver header and source files.

### 3 .\SampleCode\

ISP\	Standard ISP bootloader source code for ISP programmer.
PowerManagement\	The power consumption setting samples.
RegBased\	Demonstrate the usage of MG51 series MCU peripheral driver.
Template\	A project template for MG51 series MCU.

## 4 .\SampleCode\RegBased

<b>ADC_Bandgap_VDD</b>	Calculate the real $V_{DD}$ value of the device system based on the difference between the pre-stored ADC conversion result values when $V_{DD}$ is 3.072V and the system converted band-gap value.
<b>ADC_GPIO_Trig</b>	Demonstrate how to use GPIO to start ADC initial setting and show the conversion result in ADCRH and ADCRL register.
<b>ADC_Multi_channel</b>	Demonstrate how to regularly sample from different ADC input channel.
<b>ADC_PWM_Trig</b>	Demonstrate how to use each of PWM timer period timeout to trigger ADC conversion.
<b>ADC_Simple</b>	Start ADC conversion by triggering ADCS bit [ADCCON0.6] and check the flag register ADCF bit [ADCCON0.7] to confirm if a conversion is finished.
<b>Fsys_ModifyHIRC</b>	Call the library file "sys.c" to modify system setting as 16 MHz or 24 MHz and check clock out pin to confirm if Fsys is modified. The MG51 HIRC can be selected within 16 MHz or 24 MHz.
<b>Fsys_Select_ECLK</b>	Change the MG51 system clock from HIRC to external clock initial setting. The MG51 external clock input ranges from 4 MHz to 24 MHz.
<b>GPIO_ClockOut</b>	Show the MG51 system clock and output from CLKO pin.
<b>GPIO_Input_Output</b>	Toggle each MG51 GPIO pin output from high to low after 200ms delay.
<b>I2C_EEPROM</b>	Show how to use MG51 as master to read external connect EEPROM by I <sup>2</sup> C bus.
<b>IAP_AP_program_AP_Dataflash</b>	Demonstrate how MG51 APROM is used as Data Flash to implement erase / program / read verify function. All APROM memory can be used as Data Flash.
<b>IAP_AP_program_LD</b>	Demonstrate how MG51 IAP runs in APROM to program LDROM and implement erase / program / read verify function. User first needs to confirm if the LDROM is enabled through CONFIG setting.
<b>IAP_Dataflash_EEPROM</b>	Simulate Data Flash as EEPROM mode by calling the library file "eeprom.c". This process includes copy one page of

<b>Mode</b>	Data Flash values in RAM, modify data, erase Data Flash, then copy new values from RAM to Data Flash.
<b>IAP_LD_Program_AP</b>	Demonstrate how MG51 IAP runs in LDROM to program APROM and implement erase / program / read verify function.
<b>IAP_program_Config</b>	Demonstrate using MG51 IAP command to modify CONFIG bytes.
<b>IAP_Read_UCID</b>	Demonstrate using MG51 IAP command to read the unique customer ID (UCID). Only for customer special order MG51 MCU. One UCID is only for one customer.
<b>IAP_Read_UID</b>	Demonstrate using MG51 IAP command to read the Unique code of MG51. The UID value of each MG51 is different.
<b>INT0_ExtInt</b>	Perform MG51 external interrupt pin INT0 enabled initial setting.
<b>INT1_ExtInt</b>	Perform MG51 external interrupt pin INT1 enabled initial setting.
<b>Interrupt_ISR_all</b>	List all interrupt sector in the library file "isr.c". Call this file to implement interrupt subroutine.
<b>Pin_Interrupt</b>	Demonstrate how to wake up MG51 from Idle / Power-down mode through external interrupt input by enabling MG51 pin interrupt function.
<b>PWM0_DeadTime</b>	Configure PWM as Complementary mode. Control 3 pairs output, set each 2 channel PWM output as same duty and insert dead time.
<b>PWM0_Independent_Reload</b>	Configure PWM as independent mode. Each PWM channel outputs independently and each PWM channels output with different duty and interrupt enabled.
<b>PWM0_Synchronous</b>	Configure PWM as Synchronous mode. Each PWM0 channel 0/2/4 outputs different duty and PWM0 channel 1/3/5 duty following 0/2/4 setting.
<b>SPI_Flash</b>	Connect MG51 with W25Q16BV SPI Flash and set it as master to read and write data sample code.
<b>Timer0_mode_0_Interrupt</b>	Configure Timer 0 as mode 0, 13-bit timer with interrupt enabled initial setting.
<b>Timer0_mode_1_Interrupt</b>	Configure Timer 0 as mode 1, 16-bit timer with interrupt enabled initial setting.

<b>Timer0_mode_2_Interrupt</b>	Configure Timer 0 as mode 2, 8-bit timer with auto reload function and interrupt enabled initial setting.
<b>Timer01_mode_3_Interrupt</b>	Configure Timer 0 mode 3 as two separated timer initial settings with interrupt enabled. Timer0 counter value register high byte TH0 overflow will set the register flag bit TF1 as 1 (Timer1 overflow flag).
<b>Timer1_mode_0_Interrupt</b>	Configure Timer 1 as mode 0, 13-bit timer with interrupt enabled initial setting.
<b>Timer1_mode_1_Interrupt</b>	Configure Timer 1 as mode 1, 16-bit timer with interrupt enabled initial setting.
<b>Timer1_mode_2_Interrupt</b>	Configure Timer 1 as mode 2, 8-bit timer with auto reload function and interrupt enabled initial setting.
<b>Timer2_AutoReload_Capture</b>	Configure Timer 2 as one channel input capture with interrupt enabled initial setting. Timer 2 capture interrupt vector is different to the Timer 2 overflow interrupt.
<b>Timer2_AutoReload_Delay</b>	Configure Timer 2 as auto reload delay setting with interrupt enabled initial setting.
<b>Timer3</b>	Configure Timer 3 as auto reload mode initial setting and interrupt enabled.
<b>UART0_Interrupt_RW</b>	Configure UART0 transfer including transmit and receive with interrupt enabled.
<b>UART0_Printf</b>	Loop transmit from UART0 TXD pin initial setting with printf function API.
<b>UART1_Interrupt_RW</b>	Configure UART1 transmit and receive initial setting and enable interrupt subroutine.
<b>WakeupTimer_Interrupt</b>	Enable wake-up timer with interrupt function. Main loop enters Power-down mode after initial setting, and once WKT timeout, MG51 will wake up and then jump into interrupt subroutine to toggle GPIO output.
<b>Watchdog_Interrupt</b>	Demonstrate Watchdog Timer (WDT) initial setting with interrupt enabled and Watchdog Timer reset function disabled. The WDT counter overflow will jump into WDT interrupt subroutine.
<b>Watchdog_Reset</b>	Demonstrate Watchdog Timer (WDT) initial setting with reset function enabled in CONFIG.



Based on the features of the different products these projects is not necessarily included in folder ..\SampleCode\RegBased .

## 1 REVISION HISTORY

Date	Revision	Description
2023.05.23	1.00	Initial release.

### **Important Notice**

**Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".**

**Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.**

**All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.**

---

*Please note that all data and specifications are subject to change without notice.  
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*